

INFO0054 - Programmation fonctionnelle

Répétition 6: Arbres

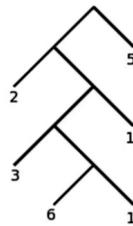
Jean-Michel BEGON

27 Mars 2018

Les arbres binaires complets

Exercice 1.

Spécifier une représentation des arbres binaires complets (chaque nœud a exactement 0 ou 2 fils) dont seules les feuilles sont étiquetées. Par exemple :



Comment adapter cette spécification si les nœuds internes sont étiquetés également ?

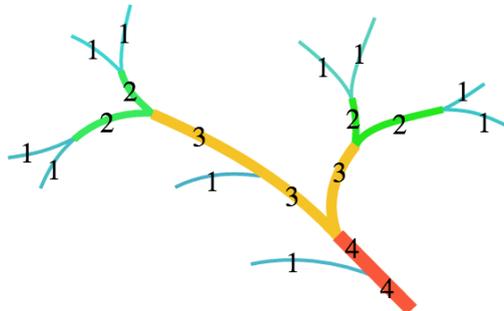
Représenter l'arbre de la figure précédente dans les deux situations. On prendra comme étiquette d'un nœud interne la somme des étiquettes de ses fils.

Exercice 2.

Le nombre de Strahler $S(t)$ d'un arbre binaire complet t est défini comme suit :

$$S(t) = \begin{cases} 1, & \text{si } t \text{ est une feuille.} \\ \max\{S(l), S(r)\}, & \text{si } t \text{ n'est pas une feuille et } S(l) \neq S(r). \\ S(l) + 1, & \text{si } t \text{ n'est pas une feuille et que } S(l) = S(r). \end{cases}$$

où l et r sont respectivement les fils gauche et droit de t .



Ecrire une fonction `strahler` prenant un arbre binaire complet t dont seules les feuilles sont étiquetées et renvoyant le nombre de Strahler de cette arbre.

Exercice 3.

Soit t un arbre binaire complet dont seules les feuilles sont étiquetées. Simplifier t consiste à supprimer dans cet arbre les feuilles redondantes. Ainsi tout nœud ayant deux fils étiquetés par le même symbole est remplacé par ce symbole et ainsi de suite.

Écrire une fonction `simplify` qui prend pour argument un arbre binaire complet t et qui retourne l'arbre binaire complet simplifié correspondant.

Exercice 4.

Un arbre arithmétique est un arbre binaire complet dont les nœuds internes sont étiquetés par l'un des symboles `add`, `sub`, `mul` et `div` et dont les feuilles sont étiquetées par un nombre.

Spécifier une représentation pour les arbres arithmétiques et écrire la fonction `value` qui prend comme argument un arbre arithmétique et qui renvoie la valeur de l'expression arithmétique associée. Si l'évaluation de celle-ci implique une division par 0, renvoyer `#f`.

`(value '(mul (add (add 3 5) (sub 3 4)) (div 3 2.0))) ⇒ 10.5`

Exercice 5.

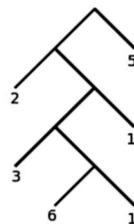
Écrire une fonction `fmap` qui prend en entrée une fonction f de $\mathbb{R} \rightarrow \mathbb{R}$ et un arbre binaire complet dont seules les feuilles sont étiquetées par des nombres et qui renvoie l'arbre de même structure dont toutes les étiquettes x ont été remplacées par $f(x)$.

Faire de même pour un arbre dont tous les nœuds sont étiquetés.

Exercice 6.

Écrire une fonction `depth-first` qui prend comme argument un arbre binaire complet dont uniquement les feuilles sont étiquetées, et renvoie la liste des étiquettes des feuilles, obtenue par un parcours en profondeur d'abord, et de gauche à droite, de l'arbre.

Par exemple, `depth-first` appliquée à l'arbre



renvoie la liste `(2 3 6 1 1 5)`.

Exercice 7.

Écrire une fonction `breadth-first` qui prend comme argument un arbre binaire complet dont uniquement les feuilles sont étiquetées, et renvoie la liste des étiquettes des feuilles, obtenue par un parcours en largeur d'abord, et de gauche à droite, de l'arbre.

Par exemple, `breadth-first` appliquée à l'arbre de la question précédente donnerait la liste `(5 2 1 3 6 1)`.

Les arbres n-aires

Exercice 8.

Spécifier la représentation d'un arbre n-aire dont tous les nœuds sont étiquetés par des atomes.

Exercice 9.

Réécrire la fonction `simplify` dans le cas d'arbres n-aires dont seules les feuilles sont étiquetées.

Exercice 10.

Soit `t` un arbre quelconque dont les feuilles sont étiquetées par des entiers naturels. (`expand t`) renvoie l'arbre obtenu en remplaçant chaque feuille de `t` étiquetée $n > 0$ par un nœud interne dont les n fils sont des feuilles étiquetées $n - 1$. Ecrire la fonction `expand`.